Naval Research Laboratory
Stennis Space Center, MS 39529-5004

NRL/MR/7440--03-8299

A Computationally Efficient Technique for the Improvement of the Display of Geospatial Information Stored in Geographic Coordinates

Hillary C. Mesick
Frank P. McCreedy
Michael M. Harris

Mapping, Charting, and Geodesy Branch
Marine Geosciences Division

July 11, 2003

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* July 11, 2003	**2. REPORT TYPE** NRL Memorandum Report	**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE A Computationally Efficient Technique for the Improvement of the Display of Geospatial Information Stored in Geographic Coordinates		**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER** 0603704N
6. AUTHOR(S) Hillary C. Mesick, Frank P. McCreedy, and Michael M. Harris		**5d. PROJECT NUMBER** 74-7441-B3 **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory Marine Geosciences Division Stennis Space Center, MS 39529-5004		**8. PERFORMING ORGANIZATION REPORT NUMBER** NRL/MR/7440--03-8299
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) SPAWAR PMW 155 4301 Pacific Highway San Diego, CA 92110		**10. SPONSOR / MONITOR'S ACRONYM(S)** SPAWAR **11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Geographical Information Systems (GIS) frequently store positional information in geographic coordinates, i.e., degrees of latitude and longitude. As a result, when GIS data are displayed on a video terminal, it is a usual practice to display the information "un-projected" with the view window x and y axis scaled in decimal degrees with degrees of longitude and latitude having the same scale factor on each axis. While this practice results in fast display time, avoiding the computational load imposed by complex cartographic projections, it results in a display that distorts the spatial relationships of the elements displayed on screen unless the displayed area is near the equator. A simple method is proposed as an alternative that greatly improves the display "fidelity" without adding any significant additional computational load.

15. SUBJECT TERMS

Positional information; Geographic information

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Hillary C. Mesick
Unclassified	Unclassified	Unclassified	UL	14	**19b. TELEPHONE NUMBER** *(include area code)* (228) 688-5257

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Contents

A COMPUTATIONALLY EFFICIENT TECHNIQUE FOR THE IMPROVEMENT OF THE DISPLAY OF GEOSPATIAL INFORMATION STORED IN GEOGRAPHIC COORDINATES

1.0 Introduction

Geographic Information Systems (GIS) frequently store positional information in geographic coordinates, i.e. degrees of latitude and longitude. As a result, when GIS data is displayed on a video terminal it is a usual practice to display the information "un-projected" with the view window x and y axis scaled in decimal degrees with degrees of longitude and latitude having the same scale factor on each axis. While this practice results in fast display time, avoiding the computational load imposed by complex cartographic projections, it results in a display that distorts the spatial relationships of the elements displayed on screen unless the displayed area is near the equator. A simple method is proposed as an alternative that greatly improves the display "fidelity" without adding any significant additional computational load.

2.0 The Earth is not Flat

When both the x and y axis employ the same linear scale factor for the display of data in geographic coordinates, the implicit assumption is made that the linear distance covered on the earth by a degree of latitude or longitude is the same. This type of cartographic projection is given the name Platte Carree and is one of the earliest projections ever used. While this assumption of equal distance for degrees of longitude and latitude produces small distortion near the equator, the distortion rapidly increases as one moves away from the equator and towards the poles. Lines of longitude, or meridians, converge as one moves from the equator toward the poles. For a spherical earth, the shortening of a degree of longitude would be proportional to the cosine of the latitude. For example, if we assume that a degree of latitude and a degree of longitude on the earth cover equal distances at the equator, at 30 degrees north the distance covered by a degree of longitude would have shortened by a factor of $\cos(30) = .86$, at 60 degrees $\cos(60) = .5$, etc. Thus at 60 degrees north, when displaying a square on the earth surface it would appear as rectangle twice as wide as it is tall. Similarly, we have also turned circles into ellipses, and perpendicular intersecting lines (other than north/south east/west) into lines that no longer intersect at right angles. Figures 1 and 2 illustrate these distortions. These two figures are of the SuperDome and the French Quarter in New Orleans, Louisiana. It is easy to see that the SuperDome is not displayed round and the street intersections in the French Quarter are not displayed perpendicular. Note that these images were derived from USGS Digital Ortho-Photo Quads that by nature have the same linear scale factor in both axis but become distorted when viewed in the Platte Carree projection (also frequently called Geographic projection). The same distortions would also be apparent with vector data for the same locations.

Manuscript approved June 9, 2003.

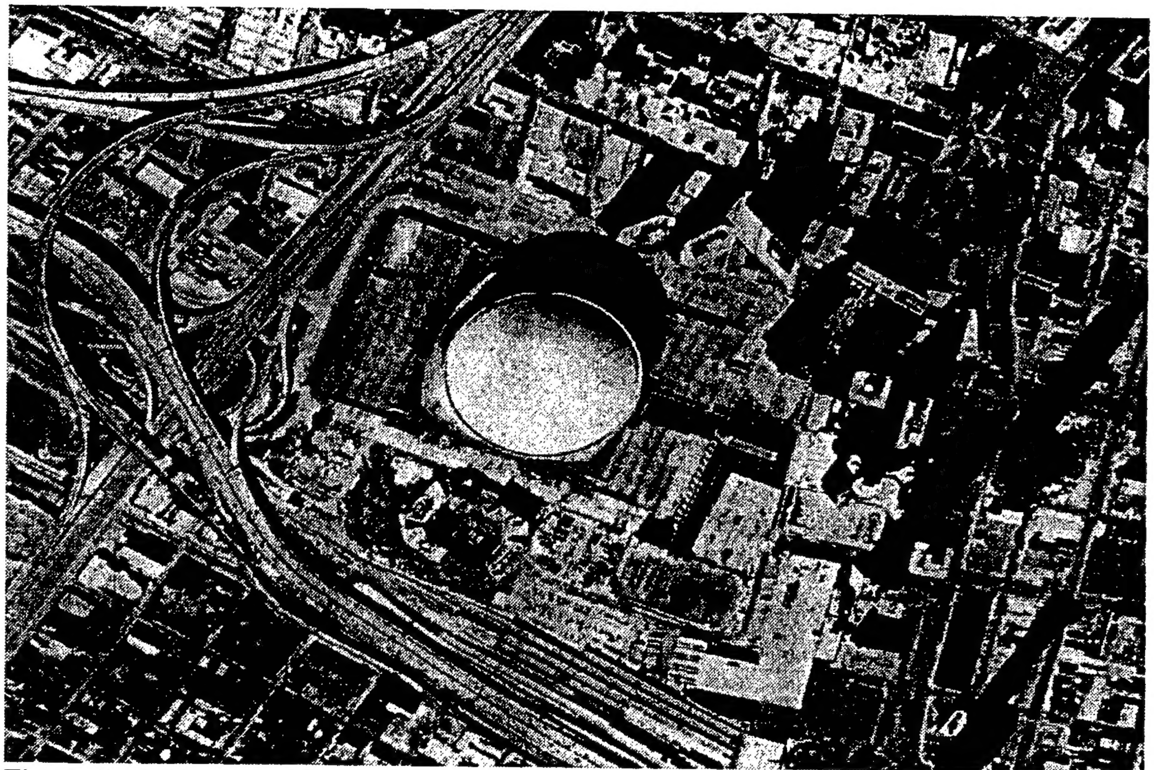


Fig. 1. SuperDome


Fig. 2. French Quarter

Rather than assuming that a degree of longitude is equal to a degree of latitude in spatial extent as is frequently done, we employ a more accurate measure of the linear

relationships in x and y and adjust the display window extent in latitude and longitude to give an approximate 1:1 spatial aspect ratio.

3.0 The Earth isn't round, it's Elliptical

For most applications, the figure of the earth is approximated by an ellipsoid of revolution that best fits the geoid (an eqi-potential surface that approximates mean sea level). Several ellipsoidal approximations are currently in use, but for our purposes we will confine our discussion to the WGS-84 ellipsoid. Any ellipse may be specified by two numbers, the semi-major axis length and the semi-minor axis length, frequently symbolized by "a" and "b". Other mathematical relationships often used in the specification of an ellipse are flattening, "f", and eccentricity "e". It is easy to see that any two of these are sufficient to define an ellipse.

$$a = SemiMajorAxis$$
$$b = SemiMinorAxis$$

$$f = (a-b)/a$$
$$e = \sqrt{a^2 - b^2}/a$$
$$e^2 = a^2 - b^2/a^2$$

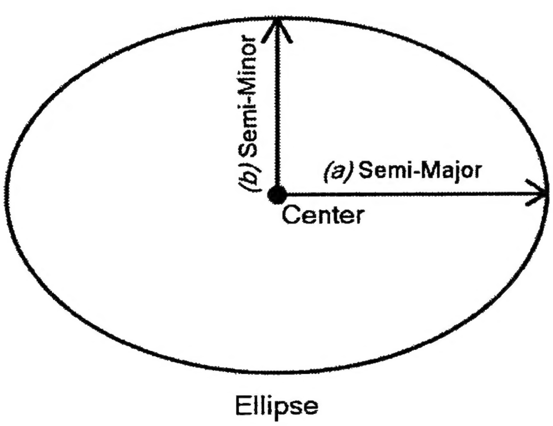
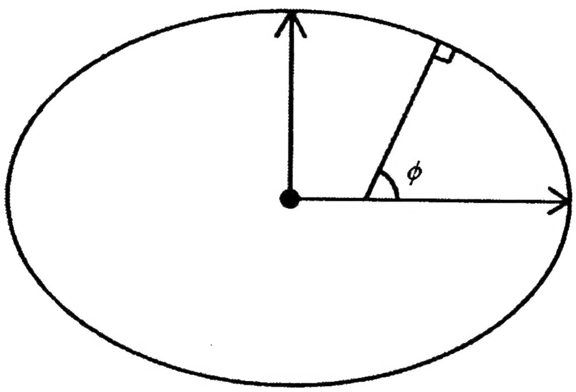


Fig. 3. General Ellipse

The WGS-84 ellipsoid is defined by the values a = 6,378,137 meters and f = 1 / 298.257223563 from which we could calculate the value of b = 6,356,752.3. We will subsequently use these values to determine the length of a degree of latitude and longitude on the reference ellipsoid. Latitude in the WGS-84 horizontal datum is the measure of the angle which is formed by the intersection of a vector normal to the

ellipsoid at the point of reference with the equatorial plane shown in Figure 4. It is not geo-centric latitude.



Geodetic Latitude ϕ

Fig. 4. Geodetic Latitude

The mathematics becomes more complicated now that we are dealing with an ellipsoid and not a sphere. In order to calculate arc length on the ellipsoid we first need to find the radius of curvature in both the meridian plane (n/s), Rm, and in the plane of the prime vertical (e/w), Rn.

With a bit of algebra and calculus the equations for the determination of these two radii can be derived. The interested reader should consult Ewing and Mitchell[1] for a development of the required derivations.

$$Rn = \frac{a}{\sqrt{1-e^2 \sin^2 \phi}}$$

$$Rm = \frac{a\left(1-e^2\right)}{\left(1-e^2 \sin^2 \phi\right)}$$

Where a and e are as defined earlier for the particular ellipsoid of choice and ϕ is the geodetic latitude.

Once we have determined the two radii of curvature we can proceed to calculate arc length. The length of arc along a parallel of latitude is straightforward since the radius of curvature Rn and its projection in the equatorial plane is not a function of longitude λ.

[1] Reference Bibliography Item # 1

(See Fig. 5.) Thus we have the length of arc along a parallel of latitude for any change in longitude as follows:

$$L_{\lambda} = Rn \cos\phi(\lambda_2 - \lambda_1)$$

where the change in longitude is measured in radians.

However the length of arc along a meridian is more complex, given by

$$L_{\phi} = \int_{\phi 1}^{\phi 2} Rm \, d\phi$$

Since this is an elliptical integral that cannot be solved in closed form we have two choices of solution. The first uses numerical integration and the second expands the function Rm as a series approximation and integrates each term separately.

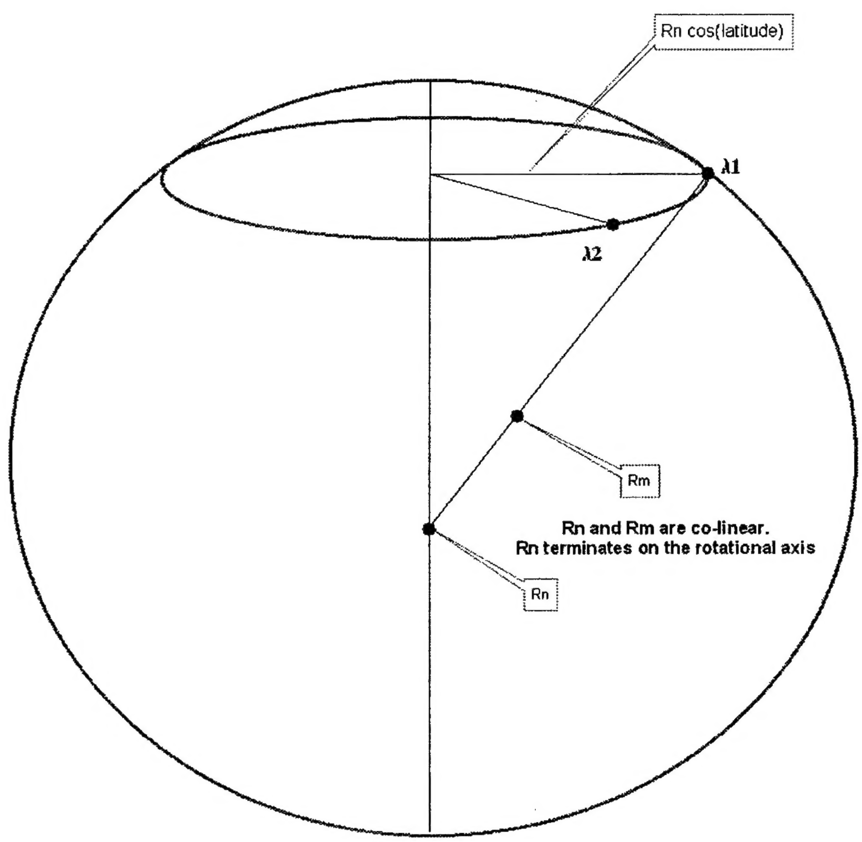


Fig. 5. Relationship of Rn and Rm to the ellipsoid and calculated values

3.1 A Simplification

Two formulas[2] and an assortment of constants for the WGS-84 ellipsoid are given that approximate the lengths of one degree of longitude and latitude for any given latitude. These formulas are accurate to well within one meter and should be entirely adequate for the stated purpose of improving GIS display as opposed to geodetic survey applications.

$$latlength = m1 + (m2\cos 2\phi) + (m3\cos 4\phi) + (m4\cos 6\phi)$$
$$lonlength = p1\cos\phi + (p2\cos 3\phi) + (p3\cos 5\phi)$$

where

$$m1 = 111132.92$$
$$m2 = -559.82$$
$$m3 = 1.175$$
$$m4 = -0.0023$$
$$p1 = 111412.84$$
$$p2 = -93.5$$
$$p3 = 0.118$$

In the use of these equations, the reader may find that for his/her particular application the smaller terms in the sums may supply more accuracy than required, and could be discarded in favor of computational efficiency. The reader may also wish to verify the results of the implementation by checking the Java calculator applet at the National Imagery and Mapping Agency (NIMA) web page address: http://164.214.12.145/calc/calc_options.html#Conversions Calculator[3]

4.0 Implementation

Frequently the GIS user, by means of graphically indicating an Area of Interest (AOI) designates the spatial extent of a display. A bounding box determines this AOI with the extents given in degrees of Latitude and Longitude. The central latitude of the AOI is easily determined by the arithmetic mean of the bounding latitudes. This central latitude value is then used to calculate both the length of a degree of longitude and the length of a degree of latitude.

---

[2] Formulas and constants taken from the NIMA JAVA applet located at http://164.214.12.145/calc/calc_options.html#Conversions Calculator

[3] Reference Bibliography Item #2

The ratio of these two values is then used to symmetrically adjust the AOI height or width in order to establish a 1:1 aspect ratio in "earth" extent. The result of this adjustment in AOI and its effect on display distortion is illustrated in Figures 6 and 7.

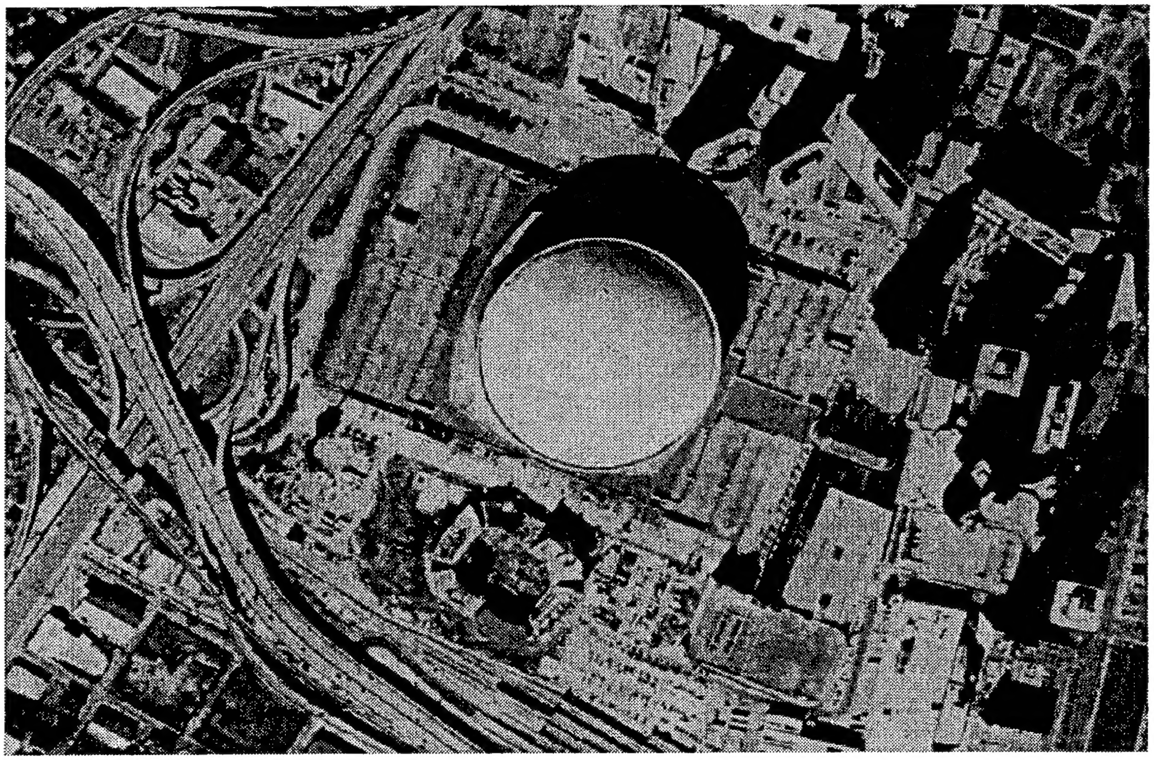


Fig. 6. SuperDome

Fig. 7. French Quarter.

These are the same geographical areas that were illustrated in Figures 1 and 2. The reader will note that the Super Dome is now approximately circular and that the street intersections of the French Quarter are perpendicular.

Additional information is supplied in the appendix that expresses the implementation technique more thoroughly in algorithmic form.

5.0 Conclusion

Calculating the ratio of lengths of a degree of latitude and longitude at the central point of an AOI and then automatically readjusting the AOI to supply a 1:1 spatial aspect ratio achieves a substantial improvement in display fidelity achieved when working with "unprojected" data in a GIS. Additionally, this improvement is gained with negligible increase in overhead. While not a substitute for the use of standard cartographic projections for high accuracy requirements, this technique does produce acceptable results for many applications and is easily implemented by the GIS application programmer. This technique is very similar to the equi-distant cylindrical projection.[4] For small geographical regions (10km) the display will be nearly conformal and the cartographic scale factor close to 1 over the entire display. However the distortion will increase as the geographical extent of the AOI is increased, but in all cases the fidelity of the display will exceed that of the "geographically projected flat earth".

6.0 Bibliography

1. Ewing, Clair E. and Mitchell, Michael M. *Introduction to Geodesy;* Elsevier North Holland Publishibg Company; NewYork, NewYork; copyright 1970; pp 8-24.

2. National Imagery and Mapping Agency (NIMA) web page address: http://164.214.12.145/calc/calc_options.html#Conversions Calculator

3. Snyder, John P. *Map Projections – A Working Manual / U.S. Geological Survey Professional Paper 1395,* U.S. Government Printing Office, Washington, 1987

[4] Reference Bibliography Item #3

7.0 Appendix

The initial area of interest is given as a bounding box (bb) which contains minimum longitude, minimum latitude, maximum longitude and maximum latitude. This initial area's longitude range to latitude range ratio matches a given map displays width to height ratio. This means that longitude and latitude scaling of the map is initially the same.

This bounding box is adjusted so that instead of longitude and latitude scaling being the same, ground distance scaling in both directions will be the same.

Conversion is handled by a method takes a bounding box and a map display width and a map display height. The method then adjusts the bounding box so that the ground distance scaling is the same in both the north-south and east-west direction. This is accomplished by JAVA

1) Determining the average (middle) latitude of the bounding box (converted to radians)

```
double averageLatitude = (((bb.maxY + bb.minY)/2.0)/360.0) * (2 * Math.PI);
```

2) Determining the kM covered in ground distance by a latitude degree at the average latitude

```
double kmInALatitudeDegree = Math.abs(111.13292 - 0.55982 * Math.cos(2.0 * averageLatitude) +
                    0.001175 * Math.cos(4.0 * averageLatitude) - 0.0000023 * Math.cos(6.0 *
averageLatitude));
```

3) Determine the kM covered in ground distance by a longitude degree at the average latitude

```
double kmInALongitudeDegree = Math.abs(111.41284 * Math.cos(averageLatitude) - 0.0935 *
Math.cos(3.0 * averageLatitude) +
                    0.000118 * Math.cos(5.0 * averageLatitude));
```

4) Calculating a map display width to map display height ratio

```
double mapAspectRatio = mapWidth / ((double)(mapHeight));
```

5) Calculating the ground distance covered by the original bounding box ratio

```
double kmCoveredInLatitude = Math.abs(kmInALatitudeDegree * (bb.maxY - bb.minY));
double kmCoveredInLongitude = Math.abs(kmInALongitudeDegree * (bb.maxX - bb.minX));
```

6) Calculating a ground distance aspect ratio for the original bounding box

```
double groundDistanceAspectRatio = kmCoveredInLongitude/kmCoveredInLatitude;
```

6) Comparing mapAspectRatio to groundDistanceAspectRatio

If mapAspectRatio is greater than groundDistanceAspectRatio then longitude range must be added to the bounding box. The new range is calculated using

```
double adjustedLongitudeRange = (mapAspectRatio * kmCoveredInLatitude)/kmInALongitudeDegree
adjustedBoundingBox.minY = bb.minY;
adjustedBoundingBox.maxY = bb.maxY;
```

adjustedBoundingBox.minX = (float)(bb.minX - Math.abs((adjustedLongitudeRange - Math.abs(bb.maxX - bb.minX))/2.0));
adjustedBoundingBox.maxX = (float)(bb.maxX + Math.abs((adjustedLongitudeRange - Math.abs(bb.maxX - bb.minX))/2.0));

8) if groundDistanceAspectRatio is greater than mapAspectRatio then latitude range must be added to the boundingbox. The new range is calculated using

double adjustedLatitudeRange = ((1/mapAspectRatio) * kmCoveredInLongitude)/kmInALatitudeDegree
adjustedBoundingBox.minX = bb.minX;
adjustedBoundingBox.maxX = bb.maxX;
adjustedBoundingBox.minY = (float)(bb.minY - Math.abs((adjustedLatitudeRange - Math.abs(bb.maxY - bb.minY))/2.0));
adjustedBoundingBox.maxY = (float)(bb.maxY + Math.abs((adjustedLatitudeRange - Math.abs(bb.maxY - bb.minY))/2.0));

9) Now the adjusted bounding box covers an area that when mapped to the map display, will show an approximately equal ground distance scaling in both the East-West and North-South directions. Accuracy will decrease as the area covered by the initial bounding box is increased.

8.0 Acknowledgments

The Oceanographer of the Navy (N096) funded this effort this effort under the direction of LCDR William Nisley (NO96) and Dr. Edward Mozley (SPAWAR) via the Space Warfare Systems Command PMW 155. This effort is part of the Naval Mapping, Charting, and Geodesy Analysis Program under Program Element 0603704N.